
(Un)interpretability of Transformers: a case study with Dyck grammars

Anonymous Authors¹

Abstract

Understanding the algorithm implemented by a model is important for trustworthiness when deploying large-scale models, which has been a topic of great interest for interpretability. In this work, we take a critical view of methods that exclusively focus on individual parts of the model, rather than consider the network as a whole. We consider a simple synthetic setup of learning a Dyck language. Theoretically, we show that the set of models that can solve this task satisfies a structural characterization derived from ideas in formal languages (the pumping lemma). We use this characterization to show that the set of optima is qualitatively rich: in particular, the attention pattern of a single layer can be “nearly randomized”, while preserving the functionality of the network. We also show via extensive experiments that these constructions are not merely a theoretical artifact: even with severe constraints to the architecture of the model, vastly different solutions can be reached via standard training. Thus, interpretability claims based on individual heads or weight matrices in the Transformer can be misleading.

1. Introduction

Transformer-based models, typically pretrained with next-token prediction objectives, serve as the basis for various applications. Being able to interpret the pretrained solutions is essential for building trustworthiness towards these models. However, certain interpretability methods can be misleading despite being highly intuitive (Jain & Wallace, 2019; Serrano & Smith, 2019; Rogers et al., 2020; Grimsley et al., 2020; Brunner et al., 2020; Meister et al., 2021).

In this work, we aim to understand the theoretical limitation of interpretability methods by characterizing the set of viable solutions. We focus on a particular toy setup in which Transformers are trained to generate *Dyck grammars*,

¹Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

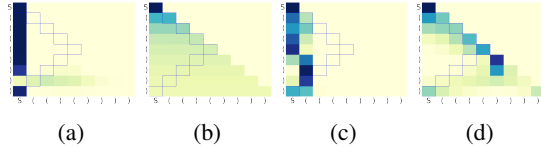


Figure 1: **Second-layer attention patterns of two-layer Transformers on Dyck** with (a,b) or without (c,d) position embedding: typical attention patterns do *not* exactly match the intuitively interpretable stack-like pattern in Ebrahimi et al. (2020); Yao et al. (2021). The blue boxes indicate the locations of the last unmatched open brackets, as they would appear in a stack-like pattern. All models reach $\geq 97\%$ accuracy (darker color indicates a higher value).

a classic type of formal language grammar consisting of balanced parentheses of multiple types. Dyck is a useful sandbox, as it captures properties like long-range dependency and hierarchical tree-like structure that commonly appear in natural and programming language syntax, and has been an object of interest in many theoretical studies (Hahn, 2020; Yao et al., 2021; Liu et al., 2022b; 2023). Dyck is canonically parsed using a stack-like data structure. Such stack-like patterns (Figure 1) have been observed in the attention heads (Ebrahimi et al., 2020; Yao et al., 2021).

Recent works (Liu et al., 2023; Li et al., 2023) show via explicit constructions of Transformer weights that Transformers are sufficiently expressive to admit very different solutions that perform equally well on the training distribution. This calls into question:

- (Q1) Do empirical solutions match the theoretical constructions given in these representational results (Figure 1)? In particular, are interpretable stack-like pattern in Ebrahimi et al. (2018) the norm or the exception?
- (Q2) More broadly, can we understand in a principled manner the fundamental obstructions to reliably “reverse engineering” the algorithm implemented by a Transformer by looking at individual attention patterns?
- (Q3) Among models that perform (near-)optimally on the training distribution, even if we cannot fully reverse engineer the algorithm implemented by the learned solutions, can we identify properties that characterize performance beyond the training distribution?

Our contributions. We provide theoretical evidence that individual components (e.g. attention patterns or weights) of a Transformer should not be expected to be interpretable.

- A **perfect balance** condition (Theorem 3.2) on the attention pattern that is sufficient and necessary for 2-layer Transformers with a *minimal first layer* (Assumption 3.1) to predict optimally on Dyck of *any* length. We then show that this condition permits abundant *non-stack-like* attention patterns that do not necessarily reflect any structure of the task, including *uniform* attentions (Corollary 3.3). We show similar results with a *near-optimal* counterpart for bounded-length Dyck (Theorem C.1).
- **Indistinguishability from a single component** (Theorem 3.4) in the sense that any Transformer can be approximated by pruning a larger random Transformer, proved via a *Lottery Ticket Hypothesis* style argument.

We further accompany these theoretical findings with an extensive set of empirical investigations.

Is standard training biased towards interpretable solutions? While both stack-like and non-stack like patterns can process Dyck theoretically, the inductive biases of the architecture or the optimization process may prefer one solution over the other in practice. In Section ??, based on a wide range of Dyck distributions and model architecture ablations, we find that Transformers that generalize near-perfectly in-distribution (and reasonably well out-of-distribution) do *not* typically produce stack-like attention patterns, showing that the results reported in prior work (Ebrahimi et al., 2018) should not be expected from standard training.

Do non-interpretable solutions perform well in practice? As a corroboration to our theory, in Section E.2, we empirically verify that we can guide Transformers to learn more balanced attention by regularizing for the balance condition, leading to better generalization.

2. Problem Setup

Dyck languages A Dyck language (Schützenberger, 1963) is generated by a context-free grammar, where the valid strings consist of balanced brackets of different types (for example, “[()]” is valid but “[()]” is not). Dyck_k denote the Dyck language defined on k types of brackets. The alphabet of Dyck_k is denoted as $\{1; 2; \dots; 2k\} \equiv [2k]$, where for each type $t \in [k]$, tokens $2t - 1$ and $2t$ are a pair of corresponding open and closed brackets. Dyck languages can be recognized by a push-down automaton. For a string w and $i \leq j \in \mathbb{Z}_+$, we use $w_{i:j}$ to denote the substring of w between position i and position j (both ends included). For a valid prefix $w_{1:i}$, the *grammar depth* of $w_{1:i}$, $\text{depth}(w_{1:i})$ is defined as the depth of the stack after processing $w_{1:i}$: $\text{depth}(w_{1:i}) = \# \text{Open Brackets in } w_{1:i} - \# \text{Closed Brackets in } w_{1:i}$.

We overload the same notation $\text{depth}(w_{1:i})$ to also denote the grammar depth of the bracket at position i . We will use $i;d$ to denote a token of type $i \in [2k]$ placed at grammar depth $d \in \mathbb{N}$. We consider bounded-depth Dyck languages following Yao et al. (2021). Specifically,

$$\text{Dyck}_{k;D} := \{w_{1:n} \in \text{Dyck}_k \mid \max_{i \in [n]} \text{depth}(w_{1:i}) \leq D\}$$

is a subset of Dyck_k such that the depth of any prefix of a word is bounded by D . While a bounded grammar depth might seem restrictive, it suffices to capture many practical settings; e.g., the level of recursion occurring in natural languages is typically bounded by a small constant (Karls-son, 2007; Jin et al., 2018). We further define the *length- N prefix set* of $\text{Dyck}_{k;D}$ as $\text{Dyck}_{k;D;N} = \{w_{1:N} \mid \exists n \geq N; w_{N+1:n} \in [2k]^{n-N}; s:t: w_{1:n} \in \text{Dyck}_{k;D}\}$. Our theoretical setup uses a fixed data distribution $\mathcal{D}_{q;k;D;N}$. Here q intuitively denotes the probability of seeing an open bracket at the next position. The formal definition is deferred to Appendix D.

Training Objectives. Given a model f parameterized by θ , we train with a *next-token prediction* language modeling objective on a given $\mathcal{D}_{q;k;D;N}$. Here the training loss is defined as $\mathcal{L}(\theta) = \mathbb{E}_{w_{1:N} \sim \mathcal{D}_{q;k;D;N}} \left[\frac{1}{N} \sum_{i=1}^N l(f(w_{1:i-1}); e_{w_i}) \right]$. For our theory analysis, we will use mean squared error as l and for experiments, we will use the cross entropy loss following common practice.

Transformer Architectures. We consider a general formulation of Transformer in this work: the l -th layer is parameterized by $\theta^{(l)} := \{W_Q^{(l)}; W_K^{(l)}; W_V^{(l)}; \text{param}(g^{(l)})\} \in \Theta$, where $W_K^{(l)}; W_Q^{(l)} \in \mathbb{R}^{m_a \times m}$, and $W_V^{(l)} \in \mathbb{R}^{m \times m}$ are the key, query, and value matrices of the attention module; $\text{param}(g^{(l)})$ are parameters of a feed-forward network $g^{(l)}$, consisting of fully connected layers, (optionally) Layer-Norms and residual links. Given $X \in \mathbb{R}^{d \times N}$, the matrix of d -dimensional features on a length- N sequence, the l -th layer of a Transformer computes

$$f_l(X; \theta^{(l)}) = g^{(l)}(\text{LN}(W_V^{(l)} X \text{Attn}(X)) + X); \quad (1)$$

$$\text{with } \text{Attn}(X) = \mathcal{C} \cdot \frac{(W_K^{(l)} X)^{\triangleright} (W_Q^{(l)} X)}{\sqrt{d_a}};$$

where \cdot^{\triangleright} is the column-wise softmax operation defined as $(A)_{i,j} = \frac{\exp(A_{i,j})}{\sum_{k=1}^N \exp(A_{k,j})}$, $\text{LN}(A)_{1:m;j} = \frac{\mathcal{P}_{\triangleright} A_{1:m;j}}{\sqrt{A_{1:m;j} k_2}} + \cdot$. $\mathcal{P}_{\triangleright}$ denotes the projection orthogonal to the $\mathbf{1}^{\triangleright}$ subspace and allows for a compact way to write the mean subtraction in LayerNorm. \mathcal{C} is the causal mask matrix defined as $C_{i,j} = 1[i \leq j]$. We call $\text{Attn}(X)$ the *Attention Pattern* of the Transformer layer l . We consider single-head attentions in this work, whose simplicity further strengthens the messages in this work.

A L -layer Transformer T_L consists of L above layers, and a word embedding matrix $W_E \in \mathbb{R}^{d \times 2k}$ and a linear decoding head with weight $W_{\text{Head}} \in \mathbb{R}^{2k \times w}$ and bias $b_{\text{Head}} \in \mathbb{R}^{2k}$. Let $Z \in \mathbb{R}^{2k \times N}$ denote the one-hot embedding of a length- N sequence, then T_L computes f_Z as

$$T_L(Z) = W_{\text{Head}} f_L \left(f_1(W_E Z) + b_{\text{Head}} \right) \quad (2)$$

Further, define the nonstructural pruning as:

Definition 2.1 (Nonstructural pruning) The nonstructural pruning¹ of a Transformer refers to the type of pruning where some entries of the weight matrices are set to zero and some LayerNorms are set as the identity.

3. Theoretical Analysis

Many prior works have looked for intuitive interpretations of Transformer solutions by studying the attention patterns of particular heads or some individual components of a Transformer (Clark et al., 2019; Vig & Belinkov, 2019; Dar et al., 2022). However, we show next why this methodology can be insufficient even in simple settings. Namely, in Transformer solutions for Dyck, neither attention patterns nor individual local components are guaranteed to encode structures specific for parsing Dyck. We further argue that the converse is also insufficient: when a Transformer does produce interpretable attention patterns, there could be limitations of such interpretation as well, as discussed in Appendix B. Together, our results provide theoretical evidence that careful analyses (beyond heuristics) are required when studying interpretations from Transformer.

3.1. Interpretability Requires Inspecting More Than Attention Patterns

This section focuses on Transformers with 2 layers, which are sufficient for processing Dyck (Yao et al., 2021). We will show that even under this simplified setting, attention patterns alone are not sufficient for interpretation. In fact, we will further restrict the set of 2-layer Transformers by requiring the first-layer outputs to only depend on information necessary for processing Dyck:

Assumption 3.1 (Minimal First Layer) We consider 2-layer Transformers with a minimal first layer f_1 : let $Z \in \mathbb{R}^{2k \times N}$ denote the one-hot embeddings of inputs $t_1, \dots, t_N \in [2k]$, then the j -th column of the output $f_1(W_E Z)$ only depends on the type and depth of $t_j \in [N]$.

The Minimal First Layer is a strong condition, as it requires the first layer output to depend only on the bracket type and depth and eliminate all other information, including positions. There are multiple constructions of a minimal

¹As opposed to structural pruning which prunes some channels of weight matrices.

first layer, such as the one in (Yao et al., 2021). When working with a minimal first layer, we will not explicitly reason about its parameterization, but instead work directly with its output. Specifically, $e_{(t;d)}$ the output of $t;d$ for $t \in [2k], d \in [D]$.

Perfect Balance Condition We find that the attention patterns alone can be too flexible to be helpful, even for the restricted class of a two-layer Transformer with a minimal first layer (Assumption 3.1) and even on a language as simple as Dyck. In particular, the second-layer attention matrix $(W_K^{(2)})^T W_Q^{(2)}$ only needs to satisfy one condition:

Theorem 3.2 (Perfect Balance, informal) Consider a two-layer Transformer T using a minimal first layer with output embedding $g_{d \in [D], i \in [2k]}$. Let $g^{(2)} := f(W_Q^{(2)}; W_K^{(2)}; W_V^{(2)}; \text{param}(g^{(2)}))$ denote the second layer weights. Under some assumptions on f , there exist $e_{(i;d)} g$ and $g^{(2)}$ that minimize the mean squared error (Eqn. 11) on Dyck $_{k,D}$ for any length N , if and only if

$$(e_{(2i-1;d^{0+1})} e_{(2i;d^0)})^T (W_K^{(2)})^T W_Q^{(2)} \quad (3)$$

$$(e_{(2j_1;d_1)} e_{(2j_2;d_2)}) = 0 \quad (4)$$

Recall that $e_{(2i-1;d^{0+1})}; e_{(2i;d^0)}$ denote the first-layer outputs of a matching pair. Equation (3) says that since matching brackets do not affect future predictions, their embeddings should balance out each other. It is important to note that the perfect balance condition does not restrict much on the attention patterns. For example, even the uniform attention satisfies the condition and can solve Dyck:

Corollary 3.3. There exists a two-layer Transformer with uniform attention and without position embedding that can generate the Dyck language of arbitrary length.

Uniform attention patterns are hardly reflective of any structure of Dyck, hence Corollary 3.3 proves that attention patterns can be oblivious about the underlying task, violating the “faithfulness” criteria for an interpretation (Jain & Wallace, 2019). We will further show in Appendix B.1 that empirically, seemingly structured attention patterns may not accurately represent the inherent structure of the task.

3.2. Interpretability Requires Inspecting More Than Any Single Weight Matrix

Another line of interpretability works involves inspecting the weight matrices of the model (Li et al., 2016; Dar et al., 2022; Eldan & Li, 2023), some of which are done locally, neglecting the interplay between different parts of the model. Our next result shows from a representational perspective that isolating single weights may also be misleading:

Theorem 3.4 (Indistinguishability From a Single Component, informal) Consider a L -layer Transformer T

with embedding dimension m , width w and $g^{(k)}(x) = \text{LN} W_2^{(k)} \text{ReLU} W_1^{(k)} x + x$. Consider a polynomial larger random Transformer T_{large} , with 4L layers, embedding dimension $4m$, and width $O(\max\{m \log \frac{wmLN}{\epsilon}; wg\})$, and same architecture choice for whose weights are sampled as $W_{ij} \sim U(-1; 1)$ for every $W \in T_{\text{large}}$. Then, with probability 1 over the randomness of T_{large} , a nonstructural pruning (Definition 2.1) of T_{large} , denote T_{large}^0 , can ϵ -approximate T . That is, $\|X \cdot T_{\text{large}}^0 - T(X)\|_2 \leq \epsilon$.

Moreover, pick any $W \in T_{\text{large}}$, with probability 1, for any smaller Transformer T_1, T_2 satisfying same conditions as T , we have two pruned Transformers $T_{\text{large},1}, T_{\text{large},2}$ based on T_{large} , such that they coincide on the pruned weight of W , and $T_{\text{large},i}$ ϵ -approximate T_i , $\|X \cdot T_{\text{large},i} - T_i(X)\|_2 \leq \epsilon$.

4. Experiments: Various Dyck Solutions

Our theory in Section 3 proves the existence of abundant non-stack-like attention patterns, all of which suffice for (near-)optimal generalization of Dyck. However, could there be implicit biases in the architecture and the optimization algorithm, which would potentially make the learned attention patterns more frequently stack-like? In this section, we show there is no evidence for such implicit bias in standard training. We will also show a modified objective based on our theory can be used explicitly regularize the model towards better length generalization (Section E.2).

We empirically verify our theoretical findings that Dyck solutions can give rise to a variety of attention patterns. We use the Adam optimizer (Kingma & Ba, 2014) unless specified otherwise. We use Transformers with 21 layers, 1 head, hidden dimension 50 and word embedding dimension 50. We test the accuracy of the model by randomly generating a Dyck prefix that ends with a closing bracket, and evaluating whether the model predicts correctly the type of the last closing bracket given the rest of the prefix. Note that in this setting a correct parser should always be able to uniquely determine the correct closing bracket type (for the sequence to remain a valid Dyck sequence). We train on valid Dyck sequences with length less than 28 generated with $\tau = 0.5$, where all models are able to achieve 97% test accuracy.

Qualitative Results. As a response to (Q1), we observe that attention patterns of Transformers trained on Dyck are not always stack-like (Figure 1). In fact, the attention patterns vary even across different random initializations. Moreover, while Theorem 3.2 predicts that position encoding is not necessary for a Transformer to generate Dyck (this is verified by experiments, as Transformers with no positional encoding achieve 97% accuracy), we observe that adding the position encoding does affect the attention patterns. We

also try using the attention layer as uniform attention and verify that they can also fit the distribution almost perfectly, which is consistent with our theory.

(a) Embedding Type 1 (b) Embedding Type 1 (c) Embedding Type 2 (d) Embedding Type 3

Figure 2: Second-layer attention patterns of two-layer Transformers with a minimal first layer: (a), (b) are based on embedding Type 1 with different learning rates, where the attention patterns show much variance as Theorem 3.2 predicts. (c), (d) are based on embedding Type 2 and Type 3. Different embedding functions lead to diverse attention patterns, most of which are not stack-like.

We then experiment with two-layer Transformers with a minimal first layer. We experiment with three different types of embeddings, the exact format is shown in Appendix E.1. As one can observe from Figure 2, the attention patterns learned by Transformers exhibit large variance between different choices of architectures and learning rates. We observe that most of the attention patterns learned by the Transformer are not stack-like.

Quantitative Experiments. We now quantify the variation in attention by comparing across multiple random initializations. We define the attention variation between two attention patterns $A_1, A_2 \in \mathbb{R}^{N \times N}$ over an length N input sequence as $\text{variation}(A_1, A_2) = \frac{1}{N} \sum_{i=1}^N |A_1(i, i) - A_2(i, i)|$. We will then calculate the average variation of an architecture by running $n = 40$ random initializations and calculate the average variation between the attention patterns of the n random initializations on sequence $[[[]]]([[]])$. We will call this quantity the average attention variation.

We observe that for standard two layer training with linear position embedding, the average attention variation is 2.20. For training without position embedding, the average attention variation is 2.27. Both variation is closed to the random baseline value of 2.85³, showing that the attention head learned by different initializations indeed tend to be very different. We also experiment with Transformer with a minimal first layer and the embedding in Equation (Type 1), which reduces the average variation to 1.04. We hypothesize that the structural constraints in this setting provide sufficiently strong inductive bias that limit the variability of attention patterns.

³The random baseline is calculated by generating purely random attention patterns (from the simplex, i.e. random square matrices s.t. each row sums up to 1) and calculate the average attention variation between them.

²We use the linear positional encoding following (Yao et al.,

References

- Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics* 48(1): 207–219, March 2022. doi: 10.1162/cal.100422. URL <https://aclanthology.org/2022.cl-1.7>
- Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the Ability and Limitations of Transformers to Recognize Formal Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7096–7116, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.576. URL <https://aclanthology.org/2020.emnlp-main.576>
- Satwik Bhattamishra, Arkil Patel, and Navin Goyal. On the computational power of transformers and its implications in sequence modeling. *Proceedings of the 24th Conference on Computational Natural Language Learning*, pp. 455–475, Online, November 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.conll-1.37. URL <https://aclanthology.org/2020.conll-1.37>
- Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Coenen, Emily Reif, Fernanda Viegas, and Martin Wattenberg. An interpretability illusion for bert. *arXiv preprint arXiv:2104.07143*, 2021.
- Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. On identifiability in transformers. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=BJg1f6EFDB>
- Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim. Thread: Circuits. *Distill*, 2020. doi: 10.23915/distill.00024. <https://distill.pub/2020/circuits>.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT’s attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276–286, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828. URL <https://aclanthology.org/W19-4828>
- Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. Analyzing transformers in embedding space, 2022.
- Yichuan Deng, Zhihang Li, and Zhao Song. Attention scheme inspired softmax regression, 2023.
- Javid Ebrahimi, Daniel Lowd, and Dejing Dou. On adversarial examples for character-level neural machine translation. In *International Conference on Computational Linguistics (COLING)*, 2018.
- Javid Ebrahimi, Dhruv Gelda, and Wei Zhang. How can self-attention networks recognize Dyck-n languages? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4301–4306, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.384. URL <https://aclanthology.org/2020.findings-emnlp.384>
- Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5793–5831. PMLR, 17–23 Jul 2022. URL <https://proceedings.mlr.press/v162/edelman22a.html>
- Ronen Eldan and Yuanzhi Li. Tinstories: How small can language models be and still speak coherent english?, 2023.
- Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread* 2021. <https://Transformer-circuits.pub/2021/framework/index.html>.
- Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.
- Yeqi Gao, Sridhar Mahadevan, and Zhao Song. An over-parameterized exponential regression, 2023.
- F. Gers and J. Schmidhuber. Lstm recurrent networks learn simple context-free and context-sensitive languages. *IEEE transactions on neural networks* 12(6):1333–40, 2001.
- Christopher Grimsley, Elijah Mayeld, and Julia R.S. Bursten. Why attention is not explanation: Surgical intervention and causal reasoning about neural models. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 1780–1790, Marseille,

- France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL <https://aclanthology.org/2020.lrec-1.220>.
- Michael Hahn. Theoretical limitations of self-attention in neural sequence models. *Trans. Assoc. Comput. Linguistics* 8:156–171, 2020. doi: 10.1162/tacl-00306. URL https://doi.org/10.1162/tacl-a_00306.
- John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL <https://aclanthology.org/D19-1275>.
- John Hewitt and Christopher D Manning. A structural probe for parsing syntax in word representations. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* pp. 4129–4138, 2019.
- John Hewitt, Michael Hahn, Surya Ganguli, Percy Liang, and Christopher D Manning. Rnns can generate bounded hierarchical languages with optimal memory. *arXiv preprint arXiv:2010.07515*, 2020.
- Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. Do attention heads in bert track syntactic dependencies?, 2019.
- Sarthak Jain and Byron C. Wallace. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1357. URL <https://aclanthology.org/N19-1357>.
- Samy Jelassi, Michael Eli Sander, and Yuanzhi Li. Vision transformers provably learn spatial structure. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.) *Advances in Neural Information Processing Systems*, 2022. URL <https://openreview.net/forum?id=eMW9AkXaREI>.
- Lifeng Jin, Finale Doshi-Velez, Timothy Miller, William Schuler, and Lane Schwartz. Unsupervised grammar induction with depth-bounded pcf. *Transactions of the Association for Computational Linguistics* 6:211–224, 2018.
- Ered Karlsson. Constraints on multiple center-embedding of clauses. *Journal of Linguistics* 43(2):365–392, 2007.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7057–7075, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.574. URL <https://aclanthology.org/2020.emnlp-main.574>.
- Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4365–4374, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1445. URL <https://aclanthology.org/D19-1445>.
- Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 681–691, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1082. URL <https://aclanthology.org/N16-1082>.
- Xian Li and Hongyu Gong. Robust optimization for multilingual translation with imbalanced data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.) *Advances in Neural Information Processing Systems*, volume 34, pp. 25086–25099. Curran Associates, Inc., 2021. URL <https://proceedings.neurips.cc/paper/2021/file/d324a0cc02881779dcda44a675fdcaaa-Paper.pdf>.
- Yuchen Li and Andrej Risteski. The limitations of limited context for constituency parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* pp. 2675–2687, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.208. URL <https://aclanthology.org/2021.acl-long.208>.

- 330 Yuchen Li, Yuanzhi Li, and Andrej Risteski. How do trans- August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.17. URL <https://aclanthology.org/2021.acl-short.17>
- 331 formers learn topic structure: Towards a mechanistic
- 332 understanding, 2023.
- 333
- 334 Yongjie Lin, Yi Chern Tan, and Robert Frank. Open sesame! William Merrill. Sequential neural networks as automata. In Proceedings of the Workshop on Deep Learning and
- 335 Getting inside BERT's linguistic knowledge. Proceedings Formal Languages: Building Bridges, pp. 1–13, Flo-
- 336 ings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pp. 241–253, Florence, Italy, August 2019. Association for Computa-
- 337 tional Linguistics. doi: 10.18653/v1/W19-4825. URL <https://www.aclweb.org/anthology/W19-3901>
- 338 Florence, Italy, August 2019. Association for Computa-
- 339 tional Linguistics. doi: 10.18653/v1/W19-4825. URL <https://www.aclweb.org/anthology/W19-3901>
- 340 <https://aclanthology.org/W19-4825>
- 341
- 342 Bingbin Liu, Daniel Hsu, Pradeep Kumar Ravikumar, and Paul Michel, Omer Levy, and Graham Neubig. Are
- 343 Andrej Risteski. Masked prediction: A parameter identi- sixteen heads really better than one? In H. Wallach,
- 344 ability view. In Alice H. Oh, Alekh Agarwal, Danielle H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox,
- 345 Belgrave, and Kyunghyun Cho (eds.), Advances in Neural and R. Garnett (eds.), Advances in Neural Information
- 346 Information Processing Systems 2022a. URL <https://openreview.net/forum?id=Hbv1b4D1aFC> Processing Systems, volume 32. Curran Associates, Inc.,
- 347 <https://openreview.net/forum?id=Hbv1b4D1aFC> 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/2c601ad9d2ff9bc8b282670cdd54f69f-Paper.pdf
- 348 Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krish- Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith,
- 349 namurthy, and Cyril Zhang. Transformers learn short- and Jacob Steinhardt. Progress measures for grokking
- 350 cuts to automata. In The Eleventh International Confer- via mechanistic interpretability. In The Eleventh In-
- 351 ence on Learning Representations 2023. URL <https://openreview.net/forum?id=De4FYqjFueZ> ternational Conference on Learning Representations
- 352 <https://openreview.net/forum?id=De4FYqjFueZ> 2023. URL <https://openreview.net/forum?id=De4FYqjFueZ>
- 353
- 354 Hong Liu, Sang Michael Xie, Zhiyuan Li, and Tengyu 2023. URL <https://openreview.net/forum?id=9XFSbDPmdW>
- 355 Ma. Same pre-training loss, better downstream: Im- id=9XFSbDPmdW
- 356 plicit bias matters for language models. arXiv preprint
- 357 arXiv:2210.141992022b.
- 358
- 359 Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, Toan Q. Nguyen and Julian Salazar. Transformers with-
- 360 and Jiawei Han. Understanding the difficulty of training- out tears: Improving the normalization of self-attention.
- 361 ing transformers. In Proceedings of the 16th International Conference on Spoken Language Translation, Hong Kong, Novem-
- 362 ence on Empirical Methods in Natural Language Process- ber 2-3 2019. Association for Computational Linguistics. URL <https://aclanthology.org/2019.iwslt-1.17>
- 363 (EMNLP), pp. 5747–5763, Online, November
- 364 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.463. URL <https://aclanthology.org/2020.emnlp-main.463>
- 365 <https://aclanthology.org/2020.emnlp-main.463>
- 366 <https://aclanthology.org/2020.emnlp-main.463>
- 367
- 368 Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas
- 369 Peters, and Noah A. Smith. Linguistic knowledge and Joseph, Nova DasSarma, Tom Henighan, Ben Mann,
- 370 and transferability of contextual representations. Pro- Amanda Askeell, Yuntao Bai, Anna Chen, Tom Con-
- 371 ceedings of the 2019 Conference of the North American erly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds,
- 372 Chapter of the Association for Computational Linguis- Danny Hernandez, Scott Johnston, Andy Jones, Jack-
- 373 tics: Human Language Technologies, Volume 1 (Long son Kernion, Liane Lovitt, Kamal Ndousse, Dario
- 374 and Short Papers), pp. 1073–1094, Minneapolis, Min- Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam
- 375 nesota, June 2019. Association for Computational Lin- McCandlish, and Chris Olah. In-context learning
- 376 guistics. doi: 10.18653/v1/N19-1112. URL <https://aclanthology.org/N19-1112> and induction heads. Transformer Circuits Thread
- 377 <https://aclanthology.org/N19-1112> 2022. <https://Transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html>
- 378
- 379 Clara Meister, Stefan Lazov, Isabelle Augenstein, and Ryan Ankit Pensia, Shashank Rajput, Alliot Nagle, Harit
- 380 Cotterell. Is sparse attention more interpretable? In Vishwakarma, and Dimitris Papailiopoulos. Opti-
- 381 Proceedings of the 59th Annual Meeting of the Asso- mal lottery tickets via subset sum: Logarithmic over-
- 382 ciation for Computational Linguistics and the 11th In- parameterization is sufficient. Advances in neural infor-
- 383 ternational Joint Conference on Natural Language Pro- mation processing systems, pp. 2599–2610, 2020.
- 384 cessing (Volume 2: Short Papers), pp. 122–129, Online, Jorge Perez, Pablo Barcelo, and Javier Marinkovic. Atten-
- tion is turing-complete. Journal of Machine Learning

- 385 Research 22(75):1–35, 2021. URL <http://jmlr.org/papers/v22/20-302.html>
- 386
- 387
- 388 Sai Prasanna, Anna Rogers, and Anna Rumshisky. When BERT Plays the Lottery, All Tickets Are Winning. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3208–3229, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.259. URL <https://aclanthology.org/2020.emnlp-main.259>
- 389
- 390
- 391
- 392
- 393
- 394
- 395
- 396
- 397 Alessandro Raganato and Jörg Tiedemann. An analysis of encoder representations in transformer-based machine translation. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pp. 287–297, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5431. URL <https://aclanthology.org/W18-5431>
- 398
- 399
- 400
- 401
- 402
- 403
- 404
- 405
- 406 Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. Transactions of the Association for Computational Linguistics 8:842–866, 2020.
- 407
- 408
- 409
- 410
- 411 M.P. Schützenberger. On context-free languages and push-down automata. Information and Control, 6(3):246–264, 1963. ISSN 0019-9958. doi: [https://doi.org/10.1016/S0019-9958\(63\)90306-1](https://doi.org/10.1016/S0019-9958(63)90306-1). URL <https://www.sciencedirect.com/science/article/pii/S0019995863903061>
- 412
- 413
- 414
- 415
- 416
- 417
- 418 So a Serrano and Noah A. Smith. Is attention interpretable? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2931–2951, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1282. URL <https://aclanthology.org/P19-1282>
- 419
- 420
- 421
- 422
- 423
- 424
- 425
- 426
- 427
- 428
- 429
- 430
- 431
- 432
- 433
- 434
- 435
- 436
- 437
- 438
- 439
- Mirac Suzgun, Yonatan Belinkov, Stuart Shieber, and Sebastian Gehrmann. LSTM networks can perform dynamic counting. In Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges, pp. 44–54, Florence, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3905. URL <https://www.aclweb.org/anthology/W19-3905>
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert re-discovers the classical nlp pipeline. arXiv preprint arXiv:1905.05950, 2019.
- Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pp. 63–76, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4808. URL <https://aclanthology.org/W19-4808>
- Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5797–5808, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580. URL <https://aclanthology.org/P19-1580>
- Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In International Conference on Learning Representations, 2023. URL <https://openreview.net/forum?id=NpsVSN6o4ul>
- Colin Wei, Yining Chen, and Tengyu Ma. Statistically meaningful approximation: a case study on approximating turing machines with transformers, 2021. URL <https://arxiv.org/abs/2107.13163>
- Gail Weiss, Yoav Goldberg, and Eran Yahav. On the practical computational power of finite precision rnns for language recognition. arXiv preprint arXiv:1805.04908, 2018.
- Gail Weiss, Yoav Goldberg, and Eran Yahav. Thinking like transformers. In Marina Meila and Tong Zhang (eds.), Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pp. 11080–11090. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/weiss21a.html>
- Sarah Wiegrefe and Yuval Pinter. Attention is not not explanation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and

- 440 the 9th International Joint Conference on Natural Lan-
441 guage Processing (EMNLP-IJCNLP), pp. 11–20, Hong
442 Kong, China, November 2019. Association for Compu-
443 tational Linguistics. doi: 10.18653/v1/D19-1002. URL
444 <https://aclanthology.org/D19-1002>
- 445
446 Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. Perturbed
447 masking: Parameter-free probing for analyzing and in-
448 terpreting BERT. In Proceedings of the 58th Annual
449 Meeting of the Association for Computational Linguistics
450 pp. 4166–4176, Online, July 2020. Association for Com-
451 putational Linguistics. doi: 10.18653/v1/2020.acl-main.
452 383. URL <https://aclanthology.org/2020.acl-main.383>
- 453
454 Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin
455 Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei
456 Wang, and Tie-Yan Liu. On layer normalization in the
457 transformer architecture. Proceedings of the 37th In-
458 ternational Conference on Machine Learning, ICML’20.
459 JMLR.org, 2020.
- 460
461 Shunyu Yao, Binghui Peng, Christos Papadimitriou, and
462 Karthik Narasimhan. Self-attention networks can pro-
463 cess bounded hierarchical languages. Proceedings of
464 the 59th Annual Meeting of the Association for Computa-
465 tional Linguistics and the 11th International Joint Confer-
466 ence on Natural Language Processing (Volume 1: Long
467 Papers) pp. 3770–3785, Online, August 2021. Associ-
468 ation for Computational Linguistics. doi: 10.18653/v1/
469 2021.acl-long.292. URL <https://aclanthology.org/2021.acl-long.292>
- 470
471 Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat,
472 Sashank Reddi, and Sanjiv Kumar. Are transformers uni-
473 versal approximators of sequence-to-sequence functions?
474 In International Conference on Learning Representations
475 2020. URL <https://openreview.net/forum?id=ByxRM0Ntvr>
- 476
477
478 Marvin Zhang, Henrik Marklund, Abhishek Gupta, Sergey
479 Levine, and Chelsea Finn. Adaptive risk minimization:
480 A meta-learning approach for tackling group shift
481 preprint arXiv:2007.02931, 2020.
- 482
483 Yi Zhang, Arturs Backurs, Bastien Bubeck, Ronen Eldan,
484 Suriya Gunasekar, and Tal Wagner. Unveiling transform-
485 ers with lego: a synthetic reasoning task, 2022. URL
486 <https://arxiv.org/abs/2206.04301>
- 487
488 Haoyu Zhao, Abhishek Panigrahi, Rong Ge, and Sanjeev
489 Arora. Do transformers parse while predicting the masked
490 word?, 2023.
- 491
492
493
494

Appendix

A. Related Work

There has been a flourishing line of work on interpretability in natural language processing. Multiple “probing” tasks have been designed to extract syntactic or semantic information from the learned representations (Raganato & Tiedemann, 2018; Liu et al., 2019; Hewitt & Manning, 2019; Clark et al., 2019). However, the effectiveness of probing often intricately depend on the architecture choices and task design, and sometimes may even result in misleading conclusions (Jain & Wallace, 2019; Serrano & Smith, 2019; Rogers et al., 2020; Brunner et al., 2020; Prasanna et al., 2020; Meister et al., 2021). While these challenges do not completely invalidate existing approaches (Wiegrefe & Pinter, 2019), it does highlight the need for more fundamental understanding of interpretability.

Towards this, we choose to focus on the synthetic setup of Dyck whose solution space is easier to characterize than natural languages, allowing us to identify a set of feasible solutions. While similar representational results have been studied in prior work (Yao et al., 2021; Liu et al., 2023; Zhao et al., 2023), our work emphasizes that theoretical constructions do not resemble the solutions found in practice. Moreover, the multiplicity of valid constructions suggest that understanding Transformer solutions require analyzing the optimization process, which a number of prior work has made progress on (Jelassi et al., 2022; Li et al., 2023; Deng et al., 2023).

Finally, it is worth noting that the challenges highlighted in our work do not contradict the line of prior works that aim to improve mechanistic interpretability into a trained model or the training process (Elhage et al., 2021; Olsson et al., 2022; Nanda et al., 2023; Li et al., 2023), which aim to develop circuit-level understanding of a particular model or the training process.

Interpreting Transformer solutions Prior empirical works show that Transformers trained on natural language data can produce representations that contain rich syntactic and semantic information, by designing a wide range of “probing” tasks (Raganato & Tiedemann, 2018; Liu et al., 2019; Hewitt & Manning, 2019; Clark et al., 2019; Tenney et al., 2019; Hewitt & Liang, 2019; Kovaleva et al., 2019; Lin et al., 2019; Wu et al., 2020; Belinkov, 2022) (or other approaches using the attention weights or parameters in neurons directly Vig & Belinkov, 2019; Htut et al., 2019; Sun & Marzouk, 2021; Eldan & Li, 2023). However, there is no canonical way to probe the model, partially due to the huge design space of probing tasks, and even a slight change in the setup may lead to very different (sometimes even seemingly contradictory) interpretations of the result (Hewitt & Liang, 2019). In this work, we tackle such ambiguity through a different perspective—by developing formal (theoretical) understanding of solutions learned by Transformers. Our results imply that it may be challenging to try to interpret Transformer solutions based on individual parameters (Li et al., 2016; Dar et al., 2022), or based on constructive proofs (unless the Transformer is specially trained to be aligned with a certain algorithm, as in Weiss et al., 2021).

Interpreting attention patterns Prior works (Jain & Wallace, 2019; Serrano & Smith, 2019; Rogers et al., 2020; Grimsley et al., 2020; Brunner et al., 2020; Prasanna et al., 2020; Meister et al., 2021; Bolukbasi et al., 2021a) present negative results on deriving explanations from attention weights using approaches by Vig & Belinkov (2019); Kobayashi et al. (2020; inter alia). However, Wiegrefe & Pinter (2019) argues to the contrary by pointing out flaws in the experimental design and arguments of some of the prior works; they also call for theoretical analysis on the issue. Hence, a takeaway from these prior works is that expositions on explainability based on attention requires clearly defining the notion of explainability adopted (often task-specific). In our work, we restrict our main theoretical analysis to the fully defined data distribution of Dyck language (Definition D.1), and define “interpretable attention pattern” as the stack-like pattern proposed in prior theoretical (Yao et al., 2021) and empirical (Ebrahimi et al., 2020) works. These concrete settings and definitions allow us to mathematically state our results and provide theoretical reasons.

Theoretical understanding of representability Methodologically, our work joins a long line of prior works that characterize the solution of neural networks via the lens of simple synthetic data, from class results on RNN representability (Siegelmann & Sontag, 1992; Gers & Schmidhuber, 2001; Weiss et al., 2018; Suzgun et al., 2019; Merrill, 2019; Hewitt et al., 2020), to the more recent Transformer results on parity (Hahn, 2020), Dyck (Yao et al., 2021), topic model (Li et al., 2023), and formal grammars in general (Bhattamishra et al., 2020a; Li & Risteski, 2021; Zhang et al., 2022; Liu et al., 2023; Zhao et al., 2023). Our work complements prior works by showing that although representational results can be obtained via intuitive “constructive proofs” that assign values to the weight matrices, the model does not typically converge to those intuitive solutions in practice. Similar messages are conveyed in Liu et al. (2023), which presents different types of

550 constructions using different numbers of layers. In contrast, we show that there exist multiple different constructions even
551 when the number of layers is kept the same.

552 There are also theoretical results on Transformers in terms of Turing completeness (Bhattamishra et al., 2020b; Perez et al.,
553 2021), universal approximability (Yun et al., 2020), and statistical sample complexity (Wei et al., 2021; Edelman et al.,
554 2022), which are orthogonal to our work.

555
556 Transformer optimization Given multiple global optima, understanding Transformer solutions requires analyzing the
557 training dynamics. Recent works theoretically analyze the learning process of Transformers on simple data distributions,
558 e.g. when the attention weights only depend on the position information (Jelassi et al., 2022), or only depend on the
559 content (Li et al., 2023). Our work studies a syntax-motivated setting in which both content and position are critical. We
560 also highlight that Transformer solutions are very sensitive to detailed changes, such as positional encoding, layer norm,
561 sharpness regularization (Foret et al., 2020), or pre-training task (Liu et al., 2022a). On a related topic but towards different
562 goals, a series of prior works aim to improve the training process of Transformers with algorithmic insights (Nguyen
563 & Salazar, 2019; Xiong et al., 2020; Liu et al., 2020; Zhang et al., 2020; Li & Gong, 2021, et alia). An end-to-end
564 theoretical characterization of the training dynamics remains an open problem; recent works that propose useful techniques
565 towards this goal include Gao et al., 2023; Deng et al., 2023.

566
567 Mechanistic interpretability Finally, it is worth noting that the challenges highlighted in our work do not contradict the
568 line of prior works that aim to improve mechanistic interpretability into a trained model or the training process (Cammarata
569 et al., 2020; Elhage et al., 2021; Olsson et al., 2022; Nanda et al., 2023; Li et al., 2023): although we prove that components
570 (e.g. attention scores) of trained Transformers do not generally admit intuitive interpretations based on the data distribution,
571 it is still possible to develop circuit-level understanding about a particular model, or measures that closely track the training
572 process, following these prior works.

573 574 A.1. Limitations and future work.

575
576 Our results do not preclude that interpretable attention patterns can emerge in multi-head, overparameterized Transformers
577 trained on more complex data distributions. In that case, we discuss some limitations of such interpretation in Appendix B.

578
579 Interesting directions of future work include extending our theoretical results to more complex settings (in terms of both
580 architecture choice and data distribution), theoretical characterization of the learning dynamics, and more experiments in
581 controlled settings for testing the connections between the training approach, interpretability, and task performance. We
582 motivate these questions and discuss some relevant trade-offs in Appendix B.

B. Are interpretable attention patterns useful?

Our results in Section 3 and Section 4 demonstrate that Transformers are sufficiently expressive that a (near-)optimal loss on Dyck languages can be achieved by a variety of attention patterns, many of which may not be interpretable.

However, multiple prior works have shown that for multi-layer multi-head Transformers trained on natural language datasets, it is often possible to locate attention heads that produce interpretable attention patterns (Vig & Belinkov, 2019; Htut et al., 2019; Sun & Marasović, 2021). Hence, it is also illustrative to consider the “inverse question” of (Q1): when some attention heads do learn to produce attention patterns that suggest intuitive interpretations, what benefits can they bring?

We discuss this through two perspectives:

- **Reliability of interpretation:** Is the Transformer necessarily implementing a solution consistent with such interpretation based on the attention patterns? (Section B.1)
- **Usefulness for task performance:** Are those interpretable attention heads more important for the task than other uninterpretable attention heads? (Section B.2)

We present preliminary analysis on these questions, and motivate future works on the interpretability of attention patterns using rigorous theoretical analysis and carefully designed experiments.

B.1. Can interpretable attention patterns be misleading?

We show through a simple argument that interpretations based on attention patterns can sometimes be misleading, as we formalize in the following proposition:

Proposition B.1. Consider an L -layer Transformer (Equation (2)). For any $W_K^{(l)}; W_Q^{(l)} \in \mathbb{R}^{m_a \times m} \ (1 \leq l \leq L)$, there exist $W_{\text{Head}} \in \mathbb{R}^{2k \times w}$ and $b_{\text{Head}} \in \mathbb{R}^{2k}$ such that $\Gamma(Z) = 0; \delta Z$.

While its proof is trivial (simply setting $W_{\text{Head}} = 0$ and $b_{\text{Head}} = 0$ suffices), Proposition B.1 implies that the solution represented by the Transformer could possibly be independent of the attention patterns in all the layers (l). Hence, it could be misleading to interpret Transformer solutions solely based on these attention patterns.

Empirically, Transformers trained on Dyck indeed sometimes produce misleading attention patterns.

We present one representative example in Figure 3, and Figure 4, in which interpretable attention patterns are misleading

We also present additional results in Figure 5, in which some interpretable attention patterns are misleading, and some are not

Figure 3: Even interpretable attention patterns can be misleading. For a 4-layer Transformer trained on Dyck with the copying task (with > 96% validation accuracy), i.e. the output should be exactly the same as the input, the attention patterns in some layers seem interpretable: (layer 2) attending to bracket type a) or (b); (layer 3) attending to closing brackets; (layer 4) never attending to bracket type a); However, none of them are informative of the copying task. This is possible because Transformers can use the residual connections (or weights MLPs or the value matrices) to solve copying, bypassing the need of using attention.

Similar message has been conveyed in prior works (Bolukbasi et al., 2021), and future works may aim to achieve the faithfulness, completeness, and minimality conditions in (Wang et al., 2023).

660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714

Figure 4: Even interpretable attention patterns can be misleading. For a 1-layer Transformer trained on Dyck with the copying task (with > 90% validation accuracy), i.e. the output should be exactly the same as the input, the attention pattern seems to be attending to closing brackets only, but that is not informative of the copying task.

(a) layer 1 of 4

(b) layer 3 of 4

Figure 5: Even interpretable attention patterns can be misleading. For a 4-layer Transformer trained on Dyck with the copying task (with > 96% validation accuracy), i.e. the output should be exactly the same as the input, both types of attention patterns are common: (a) attending to closing brackets, which is uninformative of the copying task; (b) attending to the current position, which solves the copying task.

B.2. Can interpretable attention patterns be important?

Kovaleva et al. (2019) observes that, when the “importance” of an attention head is defined as the performance drop the model suffers when the head is disabled, then for most tasks they test, the most important attention head in each layer not tend to be interpretable.

However, experiments by Voita et al. (2019) led to a seemingly contradictory observation: when attention heads are systematically pruned by re-tuning the Transformer with a relaxation penalty (i.e. encouraging the number of remaining attention heads to be small), most remaining attention heads that survive the pruning can be associated with certain functionalities such as positional, syntactic, or attending to rare tokens.

These works seem to bring mixed conclusions to our question: are interpretable attention heads more important for the task than other uninterpretable attention heads? We interpret these results by conjecturing that the definition of “importance” (reflected in their experimental design) plays a crucial role:

- When the importance of an attention head is defined as the performance drop when treating all other attention heads as uninterpretable, re-tuning experiments that prune/disable certain heads while keeping other heads unchanged (Michel et al., 2019; Kovaleva et al., 2019), the conclusion may be mostly pessimistic: mostly no strong connection between interpretability and importance.
- On the other hand, when the importance of an attention head is defined as the performance drop when allowing all other attention heads to adapt to its change, re-tuning experiments that jointly optimize all attention heads while penalizing the number of heads (Voita et al., 2019), the conclusion may be more optimistic: the heads obtained as a result of this optimization tend to be interpretable.

We think the following trade-offs apply:

- On one hand, the latter setting is more practical, since Transformers are typically not trained to explicitly ensure that the model performs well when a single attention head is individually disabled; rather, it would be more intuitive to think of a group of attention heads as jointly representing some transformation, so when one head is disabled, other heads should be re-tuned to adapt to the change.
- On the other hand, when all other heads change too much during such re-tuning, the resulting set of attention heads no longer admit an unambiguous one-to-one map with the original set of (unpruned) attention heads. As a result, the interpretability and importance obtained from the set of pruned heads do not necessarily imply those properties of the original heads.

A comprehensive study of this question involves multi-head extensions of our theoretical results (Section 3), and carefully-designed experiments that take the above-mentioned trade-offs into consideration. We think these directions are interesting future work.

C. Approximate Balance Condition For Finite Length Training Data

The condition in Theorem 3.2 requires the model to reach the optimal loss for data of any length. However, in practice, one can only train the model on finite length data and the model can only reach a low but non-optimal loss for finite length data. In this case, the condition in Theorem 3.2 is not precisely met. However, one can show that a similar condition is still necessary if one restricted the Lipschitz constant of the projection function. We first define two quantities that measure the deviation from the previous ideal scenario:

$$S_{d;d^0;i;j}^{(2)} = \sum_{t=2^k}^X u(2_{j;d}; 2_{i;d^0}) + u(2_{j;d}; 2_{i-1;d^0+1}) \quad (5)$$

$$t = \arg \min_{t \in [k]^d} u(2_{j;d}; 2_{t;d^0}) \quad (6)$$

$$P_{d;j}^{(2)} = \min_{t \in [k]^d; t_d^0 \in t_d} \sum_{d^0 \in d} u(2_{j;d}; 2_{t;d^0}) + u(2_{j;d}; 2_{j-1;d^0+1}) + u(2_{j;d}; 2_{j;d}) \quad (7)$$

The first term $S_{d;d^0;i;j}^{(2)}$ measures the change in the input of the LayerNorm layer for the last token when a matching pair of brackets $(2_{i;d^0}; 2_{i-1;d^0+1})$ is inserted into the prefix. Under the perfect balance condition, $S_{d;d^0;i;j}^{(2)} = 0$. The second term $P_{d;j}^{(2)}$ is measures the norm of the input of the LayerNorm layer at last token when the prefix only contains open brackets. In the following theorem, $P_{d;j}^{(2)}$ will be used as a baseline to show $S_{d;d^0;i;j}^{(2)}$ cannot be too large, i.e., the model should not be sensitive to the insertion of a matching pair of brackets.

Theorem C.1 (Approximate Balance) Consider a two-layer Transformer with a minimal first layer trained with the mean squared error (Equation 11). For any $\epsilon; N > 0$ and sufficiently small ϵ , suppose $\mathbf{g}^{(2)}$ is ϵ -Lipschitz, and suppose the set of second-layer weights $\mathbf{W}_N^{(2)}$ satisfies that $L(T[\frac{(\cdot)}{N}]; D_{q;k;D;N}) \leq \epsilon^q N$. Then, there exists a constant $C_{\epsilon;D}$, such that for any $0 \leq d^0 \leq D; 1 \leq d \leq D; i; j \in [k]$, it holds that

$$S_{d;d^0;i;j}^{(2)} \leq \frac{C_{\epsilon;D}}{N} P_{d;j}^{(2)} \quad (8)$$

Equation (8) requires $S_{d;d^0;i;j}^{(2)}$ to be small relative to $P_{d;j}^{(2)}$, and can be interpreted as a relaxation of which is equivalent to $S_{d;d^0;i;j}^{(2)} = 0$. The proof of Theorem C.1 shares similar intuition as Theorem 3.2 and is given in Appendix D.3. As a direct corollary of Theorem C.1, we can additionally consider adding a weight decay, in which case approximate balance condition holds as the regularization strength goes to 0:

Corollary C.2. Consider the setting where a Transformer with a fixed minimal first layer is trained to minimize $L(x) + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$, which is the squared loss with weight decay. Suppose the function $f^{(2)}$ of the Transformer is a fully connected network. Then, for any length N , there exists constant $C_{\lambda} > 0$, such that for parameters \mathbf{w}_N minimizing L^{reg} , it holds $0 \leq d^0 \leq D; 1 \leq d \leq D; i; j \in [k]$ that,

$$\limsup_{\lambda \rightarrow 0} \frac{S_{d;d^0;i;j}^{(2)}[\cdot; N]}{P_{d;j}^{(2)}[\cdot; N] + 1} \leq \frac{C_{\lambda}}{N}$$

D. Omitted Proofs in Section 3

D.1. Detailed Setup

Data Distribution We will first formally define the distribution we are considering.

Definition D.1 (Dyck distribution) The distribution $\mathbb{D}_{q;k;D;N}$, specified by $q \in (0; 1)$, is defined over $\text{Dyck}_{k;D;N}$ such that $\mathbb{P}(w_{1:N} \in \text{Dyck}_{k;D;N}) = 1$.

$$\mathbb{P}(w_{1:N}) = \frac{q^{\sum_{i=1}^N \mathbb{1}\{w_i = \text{open}\}} (1-q)^{\sum_{i=1}^N \mathbb{1}\{w_i = \text{closed}\}}}{\sum_{w \in \text{Dyck}_{k;D;N}} q^{\sum_{i=1}^N \mathbb{1}\{w_i = \text{open}\}} (1-q)^{\sum_{i=1}^N \mathbb{1}\{w_i = \text{closed}\}}} \quad (9)$$

That is, $q \in (0; 1)$ denote the probability of seeing an open bracket at the next position, except for two corner cases: 1) the next bracket has to be open if the current grammar depth is 0 (1 after seeing the open bracket); 2) the next bracket has to be closed if the current grammar depth is D .

Loss Function the next token prediction task used to predict the next token for any fixed pre x . Precisely, given a pre $x = w_{1:N} \in \text{Dyck}_{k;D;N}$ and a loss function $l(\cdot; \cdot) \in \mathbb{R}$, f is trained to minimize the loss function $\min_x L(x)$ for

$$L(x) = \mathbb{E}_{w_{1:N} \sim \mathbb{D}_{q;k;D;N}} \left[\frac{1}{N} \sum_{i=1}^N l(f(w_{1:i-1}); e_{w_i}) \right] \quad (10)$$

We will also consider λ_2 -regularized version $L^{\text{reg}}(x) = L(x) + \frac{\lambda}{2} \|x\|_2^2$ with parameter $\lambda > 0$.

For our theory, we will consider the mean squared error (MSE) as the loss function,

$$l := l_{\text{sq}}(x; e_i) = \|x - e_i\|_2^2 \quad (11)$$

In our experiments, we apply the cross entropy loss following common practice.

D.2. Proof of Theorem 3.2

The key step is already shown in Section 3. We will restate the proof rigorously here.

Theorem D.2 (Perfect Balance; formal version of Theorem 3.2) Consider a two-layer Transformer with a minimal first layer with output embedding $\{e_{i;d}\}_{i \in [2k], d \in [D]}$. Let $(^{(2)} := f(W_Q^{(2)}; W_K^{(2)}; W_V^{(2)}; \text{param}(g^{(2)}))$ denote the second layer weights.

Define the **balanced condition** to be the condition that for any $i_1, j_2 \in [k]$ and $d_1, d_2 \in [D]$,

$$(e_{(2i_1-1;d_1)} - e_{(2i_1;d_1)})^\top (W_K^{(2)})^\top W_Q^{(2)} (e_{(2j_2-1;d_2)} - e_{(2j_2;d_2)}) = 0 \quad (12)$$

Then, for the existence of $\{e_{i;d}\}_{i \in [2k], d \in [D]}$ and $(^{(2)}$ that achieves the Bayes-optimal loss for the mean squared error (Eqn. 11) on $\text{Dyck}_{k;D}$ for any length N , it holds that:

- If $W_V^{(2)}$ satisfies $\mathbb{P}_\gamma W_V^{(2)} e_{(t;d)} \neq 0; \forall t \in [2k]; d \in [D]$ then the balanced condition is necessary to show existence.
- Conversely, if the set of $2k$ encodings $\{e_{(2i_1-1;d_1)}; e_{(2i_1;d_1)}\}_{i_1 \in [k], d_1 \in [D]}$ are linearly independent for any $d_1 \in [D]$, then the balanced condition is sufficient to show existence.

Remark Recall that P_γ projects to the subspace orthogonal to $\mathbb{1}$. The assumption in the necessary condition can be intuitively understood as requiring all tokens to have nonzero contributions to the prediction, because otherwise $e_{(t;d)}$ will not contribute to prediction after the LayerNorm.

Proof. Necessity of the balanced condition By Equation (1), the attention output is directly used as the input of LayerNorm, thus we ignore the normalization from the softmax operation. For any prepadding with a closed bracket $e_{(2j_2;d_2)}$ for $d_2 = 1$

and containing brackets of all depths $[D]$, let p_m be the prefix obtained by inserting pairs of (\cdot, \cdot) for arbitrary $2 \leq [k]$ and depth $0 \leq [D]$. Denote the projection of the unnormalized attention output by

$$u(t_1; d_1; t_2; d_2) := P \exp \left(e_{t_1; d_1}^\top (W_K^{(2)})^\top W_Q^{(2)} e_{t_2; d_2} - W_V^{(2)} e_{t_1; d_1} \right) \quad (13)$$

Then, by Equation (2), we have,

$$T(p_m) = g^{(2)} \text{LN}^{(2)}(v + m(u(2j; d; 2i; d^0 - 1) + u(2j; d; 2i - 1; d^0))) + e(2j; d); \quad (14)$$

where v denotes the unnormalized second-layer output given input.

Towards reaching a contradiction, suppose $(e(2j; d; 2i; d^0) + u(2j; d; 2i - 1; d^0 + 1)) \notin 0$. Based on the continuity of the projection function and the LayerNorm Layer, we can show that $T(p_m)$ depend only on grammar depths d^0 and types $2j; 2i - 1; 2i$, which, however, are not sufficient to determine the next-token probability p_m since the latter depends on the type of the last unmatched open bracket. This contradicts the assumption that the model achieves the Bayes-optimal loss for any length n . Hence we must have

$$u(2j; d; 2i; d^0 - 1) + u(2j; d; 2i - 1; d^0) = 0 \quad (15)$$

Finally, since we assume $(e_{t; d}^\top W_V^{(2)} e_{t; d}) \notin 0$, we conclude that

$$(e_{2i - 1; d^0}^\top e_{2i; d^0 - 1})^\top (W_K^{(2)})^\top W_Q^{(2)} e_{2j; d} = \ln \frac{k P \sum W_V e_{2i - 1; d^0} k_2}{k P \sum W_V e_{2i; d^0 - 1} k_2} :$$

Note that the right hand side is independent of d . This concludes the proof for the necessity of the condition.

Sufficiency of the balance condition. We will show a construction, using the embedding function (\cdot, \cdot) as given in Equation (Type 1). Fix any $2 \leq [k]; d \geq 2 [D]$. By Equation (12), we can assume that there exists a $2 \leq [k]$ such that for $i \in [k], d^0 \geq 2 [D]$, it satisfies

$$a_{i; d^0}, (e_{2i - 1; d^0}^\top e_{2i; d^0 - 1})^\top (W_K^{(2)})^\top W_Q^{(2)} e_{2j; d} :$$

We can then choose $a_{i; d^0}$ for $i \in [k]$ and $d^0 \geq 2 [D]$ such that

$$\begin{aligned} W_V^{(2)} e_{2i; d^0 - 1} &= \exp(a_{i; d^0}) o_{(2i - 1)(D - 1) + d^0} \\ W_V^{(2)} e_{2i - 1; d^0} &= o_{(2i - 1)(D - 1) + d^0} \end{aligned} \quad (16)$$

Such $W_V^{(2)}$ is guaranteed to exist: solving for $W_V^{(2)}$ is equivalently to solving the linear equation $W_V^{(2)} E = O$, where $E; O \in \mathbb{R}^{2kD \times 2kD}$ are defined according to Equation (16) and E is of full rank by the linear independence assumption.

It can be checked that choosing $W_V^{(2)}$ to satisfy Equation (16) will also make Equation (15) satisfied. Hence for any prefix p of length n ending with a closed bracket (\cdot, \cdot) satisfying $d \geq 1$, suppose the list of unmatched open brackets in p is $[2j_1 - 1; 2j_2 - 1; \dots; 2j_m - 1; d]$, then suppose x is the input of the second layer, we will have the last column (i.e. corresponding to the last position) of the input to the LayerNorm satisfies,

$$W_V^{(2)} x = C \frac{(W_K^{(2)} x)^\top (W_Q^{(2)} x)}{d_a} \begin{matrix} \vdots \\ \vdots \\ \vdots \end{matrix} = \sum_{s=1}^d x^d u(2j_s - 1; s; 2j; d); \quad (17)$$

where C denotes the causal mask.

Finally we can choose the weights in the LayerNorm to be sufficiently small such that the largest index of the last column of input $\log^{(2)}$ is determined by $x_{:,n}$. This weights can always be chosen because the norm of the output of LayerNorm is bounded by 1 and $(e_{t; d})$ are linearly independent, hence nonzero. Then the next token probability can be determined by:

⁴Specifically, $E = [e_{(1;1)}; e_{(1;2)}; \dots; e_{(2k;D-2)}; e_{(2k;D-1)}]$, i.e. E is the collection of all $(t; d)$. O is defined such that for every d^0 , $O_{:, (D-1) + d^0} = \exp(a_{t=2; d^0}) o_{(t-1)(D-1) + d^0}$ if t is even, and $O_{:, (D-1) + d^0} = o_{t(D-1) + d^0}$ if t is odd.

1. The last bracket in p ends with an open bracket or a closed bracket with depth d .
2. The type of last unmatched open bracket is t . Suppose the grammar depth of this unmatched open bracket is i . We only need to look at indices $2i - 1, (D - 1) + d$ for $i \in [k]$. Among values of these indices, if the value is maximized at $i^* \in [k]$, then the correct type of the unmatched bracket is t_{i^*} .

To complete the proof, note that the above functionality can be implemented with a combination of feedforward layers. Specifically, since there are only a finite number of possible inputs, we can construct a 2-layer ReLU network that memorize the values for all inputs, which requires a width that is polynomial in the number of possible inputs. \square

D.2.1. PROOF OF COROLLARY 3.3

Corollary D.3 (Corollary 3.3, restated) There exists a two-layer Transformer with uniform attention and without position embedding (but with causal mask) that can generate the Dyck language of arbitrary length.

Proof. It is easy to see that the condition in Theorem 3.2 is satisfied. Hence it suffices to construct a uniform attention first layer that can generate the embedding in Equation (Type 1). Let $W_V^{(1)}$ be the identity matrix, and suppose z_i is the one-hot embeddings of a pre p of length n , where each token of type $t \in [2k]$ is encoded as z_t . Then, the last column of Z satisfies

$$W_V^{(1)} Z^h C = \frac{(W_K^{(1)} Z)^T (W_Q^{(1)} Z)}{d_a} \cdot \mathbf{1} = \sum_{i=1}^k \# \text{ token of type } t_i \text{ in } p \cdot z_{t_i} \quad (18)$$

where C denotes the causal mask.

The depth of the n -th token can then be determined by counting the number of indices $2i - 1$ and $2i$ in the last column of Z are different by 1. Similar to the proof of Theorem D.2, this function can be implemented with a combination of feedforward layers and LayerNorm layers and the proof is then completed. \square

D.3. Proof of Theorem C.1

Let's first define a quantity for convenience of later exposition. Let be defined as in Equation (13). For any $2 \leq k \leq D$ and $t \in [k]^{d-1}$, denote the quantity

$$Q(i; d; t) := \sum_{1 \leq d^0 < d} u(2i; d-1; 2t_{d^0-1; d^0}) + u(2i; d-1; 2i-1; d) + u(2i; d-1; 2i; d-1); \quad (19)$$

where t_{d^0} denotes the d^0 -th entry of t . That is, t is a string of $d-1$ open brackets. Let t denote a bracket of type $t \in [2k]$ without specifying the grammar depth (i.e. the grammar depth is implicit from the context). $Q(i; d; t)$ can be considered as the unnormalized output of the second-layer attention of a Transformer on the input sequence $t_{d^0-1; d^0}$.

Theorem D.4 (Approximate Balance; formal version of Theorem C.1) Consider a two-layer Transformer with a minimal first layer trained with the mean squared error (Equation (11)). For any $\epsilon; N > 0$ and sufficiently small δ , suppose $g^{(2)}$ is ϵ -Lipschitz, and suppose the set of second-layer weights satisfies that $L(T[\frac{(2)}{N}]; D_{q;k;D;N}) \leq \epsilon^N$. Then, there exists a constant $C_{\epsilon; \delta; D}$, such that for any $0 \leq d^0 \leq D; 1 \leq d \leq D; i; j \in [k]$, it holds that

$$S_{d;d^0;i;j}[\frac{(2)}{N}] \leq \frac{C_{\epsilon; \delta; D}}{N} P_{d;j}[\frac{(2)}{N}]; \quad (20)$$

where

$$S_{d;d^0;i;j}[\frac{(2)}{N}] = u(2j; d; 2i; d^0) + u(2j; d; 2i-1; d^0+1); \quad (21)$$

$$P_{d;j}[\frac{(2)}{N}] = \min_{t \in [k]^{d-1}; t_{d^0} \in t_d} kQ(i; d; t^0)k_2; \quad (22)$$

⁵ $s \cdot t$ denotes the concatenation of two strings, same as in Equation (Type 1)-(Type 3). The concatenation of two tokens is simply written as $i \cdot j$.

990 for $t = \arg \min_{t \in [k]^{d-1}} \|kQ(2j; d; t)\|_2$.⁶

991

992 Proof. The key idea is similar to the proof of necessity in Theorem 3.2. That is, we will construct two input sequences with
 993 different next-word distributions, and show that the approximate balance condition must hold so that inserting (a bounded
 994 number of) pairs of matching brackets does not collapse the two predicted distributions given by the Transformer.

995 Constructing the input sequences.
 996

997 Let $t := \arg \min_{t \in [k]^{d-1}} \|kQ(2j; d; t)\|_2$, and let t^0 denote the prefix that minimizes $\|kQ(2j; d; t)\|_2$ subject to the constraint
 998 that t^0 must differ from t in the last (i.e. $(d-1)$ th) position, i.e.

999

$$1000 \quad t^0 = \arg \min_{t \in [k]^{d-1}; t_d^0 \neq t_d} \|kQ(2j; d; t)\|_2$$

1001

1002 The motivation for such choices of t and t^0 is that since they differ at least by the last position which is an open bracket, they
 1003 must lead to different next-word distributions. Note also that $\|kQ(2j; d; t^0)\|_2 = \|kQ(2j; d; t)\|_2$.

1004 With the above definition of t and t^0 , consider two valid Dyck prefixes p_1 and p_2 with length no longer than N , defined as
 1005 follows: for any $d^0 \in [D]$; $i, j \in [k]$, consider a common prefix $p = \left| \frac{2i-1}{d^0} \{ \underbrace{\dots}_{d^0 \text{ open brackets}} \} \right| \frac{2i-1}{b \frac{N-2d^0}{2} c} \{ \underbrace{\dots}_{d^0 \text{ closed brackets}} \} \left| \frac{2i-1}{d^0} \{ \underbrace{\dots}_{d^0 \text{ closed brackets}} \} \right|$,

1006

1007 and set:
 1008

1009

$$1010 \quad p_1 = p \cdot t \cdot \frac{2j-1}{2j};$$

$$1011 \quad p_2 = p \cdot t^0 \cdot \frac{2j-1}{2j};$$

1012

1013 In the following, we will show that the approximate balance condition must hold for the predictions p_1 and p_2 to be
 1014 sufficiently different.

1015 Bounding the difference in Transformer outputs. The Transformer outputs ϕ_1 and ϕ_2 satisfies

1016

$$1017 \quad \|kT \left[\frac{2}{N} \right](p_1) - kT \left[\frac{2}{N} \right](p_2)\|_2 \leq \text{TV}(p_1; p_2) \cdot o(1) = o(1); \quad (23)$$

1018

1019 where $\text{TV}(p_1; p_2)$ denotes the TV distance in the next-word distributions for p_1 and p_2 , and $o(1)$ means the term will go
 1020 to zero for sufficiently small N . The former is bounded by the construction of p_1 and p_2 . The latter is bounded because of the
 1021 assumption on $\frac{2}{N}$, which states that the set of second-layer weights $\{T \left[\frac{2}{N} \right]; D_{q,k;D;N}\}$ is q^N -Lipschitz with
 1022 sufficiently small q .

1023

1024 Define by A_p the contribution of p to the attention output (before LayerNorm) of the last position of p , i.e.

1025

$$1026 \quad A_p = \sum_{0 \leq d^0 < d} \left(u(2j; d-1; 2i; d^0) + u(2j; d-1; 2i-1; d^0+1) \right)$$

$$1027 \quad + b \frac{N-2d^0}{2} c \left(u(2j; d-1; 2i; d^0) + u(2j; d-1; 2i-1; d^0+1) \right); \quad (24)$$

1028

1030 The attention outputs (before LayerNorm) of p_1 and p_2 , denoted by $A(p_1)$ and $A(p_2)$, satisfy that

1031

$$1032 \quad P_{\frac{2}{N}} A(p_1) = P_{\frac{2}{N}} (A_p + Q(2j; d; t));$$

$$1033 \quad P_{\frac{2}{N}} A(p_2) = P_{\frac{2}{N}} (A_p + Q(2j; d; t^0)); \quad (25)$$

1034

1035 Note that for any prefix p^0 , $T \left[\frac{2}{N} \right](p^0) = g \left(P_{\frac{2}{N}} A(p^0) \right)$. Then, since g is q^N -Lipschitz,

1036

$$1037 \quad \frac{P_{\frac{2}{N}} A(p_1)}{\|P_{\frac{2}{N}} A(p_1)\|_2} - \frac{P_{\frac{2}{N}} A(p_2)}{\|P_{\frac{2}{N}} A(p_2)\|_2} \leq \frac{1}{2} \frac{\text{TV}(p_1; p_2)}{\|P_{\frac{2}{N}} A(p_1)\|_2} = o(1); \quad (26)$$

1038

1040 We show that A_p should not be too much larger in norm than $Q(2j; d; t)$ or $Q(2j; d; t^0)$. First let's state a helper lemma
 1041 about the contrapositive:

1042

1043 ⁶Erratum: This definition of $P_{d;ij} \left[\frac{2}{N} \right]$ is slightly different from the one in the original main paper submitted on May 17th. The
 1044 definition here and in the current main paper have been corrected.

1045 Lemma D.5. For any $\epsilon > 0$, there exists a constant R , such that for any $a, b \in \mathbb{R}^d$ and any $r \in \mathbb{R}^d$ such that
 1046 $\|a\|_2 \leq R \max\{\|a\|_2, \|b\|_2\}$, it holds that

$$\frac{\|a+r\|_2}{\|a\|_2} \leq \frac{\|b+r\|_2}{\|b\|_2} + \epsilon$$

1051 Proof. Denote $r_0 := \max\{\|a\|_2, \|b\|_2\}$. Then $R := \frac{4r_0}{\epsilon} + 1$ suffices:

$$\begin{aligned} & \frac{\|a+r\|_2}{\|a\|_2} - \frac{\|b+r\|_2}{\|b\|_2} \leq \frac{\|a-b\|_2}{\|a\|_2} + \frac{\|b\|_2}{\|a\|_2} \left(\frac{\|a+r\|_2}{\|a\|_2} - \frac{\|b+r\|_2}{\|b\|_2} \right) \\ & \leq \frac{\|a-b\|_2}{\|a\|_2} + \frac{\|b\|_2}{\|a\|_2} \left(\frac{\|a-b\|_2}{\|a\|_2} + \frac{\|b\|_2}{\|a\|_2} \right) \\ & \leq \frac{\|a-b\|_2}{\|a\|_2} + \frac{\|b\|_2}{\|a\|_2} \left(\frac{\|a-b\|_2}{\|a\|_2} + \frac{\|b\|_2}{\|a\|_2} \right) \\ & \leq \frac{\|a-b\|_2}{\|a\|_2} + \frac{\|b\|_2}{\|a\|_2} \left(\frac{\|a-b\|_2}{\|a\|_2} + \frac{\|b\|_2}{\|a\|_2} \right) \\ & = \frac{\|a-b\|_2}{\|a\|_2} + \frac{\|b\|_2}{\|a\|_2} \left(\frac{\|a-b\|_2}{\|a\|_2} + \frac{\|b\|_2}{\|a\|_2} \right) \end{aligned}$$

□

1062 Lemma D.5 implies that if $\|a\|_2$ is too large, then the output $\mathbf{p}_{d,j}$ (Equation (26)) won't be sufficiently different. Let
 1063 $\mathbf{P}_{d,j} \in \mathbb{R}^{N \times N}$ be defined as in Equation (21) and let R be the constant in Lemma D.5, we need to bound $\|a\|_2$ by

$$\|a\|_2 \leq R \|\mathbf{P}_{d,j}\|_2 \tag{27}$$

1067 As Equation (27) holds for any $d \in \mathbb{D}$, by an induction on d (from 1 to D) on the second term in Equation (24), one can
 1068 show that there exists C (depending on R), such that,

$$\|S_{d;d^0;ij}\|_2 \leq C \|\mathbf{P}_{d,j}\|_2 \tag{28}$$

□

1074 Proof of Corollary C.2. This proof is in fact a direct combination of Theorems 3.2 and C.1. By Theorem 3.2 we know there
 1075 exists a weight $w^{(2)}$ that can reach zero loss for arbitrarily large N . Then it holds that $\mathbf{w}^{(2)} \in \mathbb{R}^N$ minimizes
 1076 the regularized loss. Notice bounded weight implies bounded Lipschitzness. The rest follows as Theorem C.1. □

1078 D.4. Proof of Theorem 3.4 – Indistinguishability from a single component

1080 We now show the limitation of interpretability from a single component, using a Lottery-Ticket-style argument by pruning
 1081 from large random Transformers.

1082 For this section only, we will make the following modifications to the Transformer architecture in (2):

- We lower bound the normalization factor in the LayerNorm by some constant. Namely we consider:

$$\text{LN}_C(x) = \frac{\mathbf{P}_? x}{\max\{\mathbf{P}_? x\|_2, C\}}$$

1089 We need this assumption for technical reasons (to make the LayerNorm Lipschitz). We note that thresholding is
 1090 a common practice empirically due to numerical stability concerns.

- We assume all affine layers and linear head in the Transformer have zero bias. This is mainly for technical convenience, and was also assumed in prior works on theoretical analysis of the lottery ticket hypothesis (Pensia et al., 2020). Note that this is not a restriction since bias can be removed with homogeneous coordinates.

1096 We will also consider a modified projection function $\mathbf{g}_{\text{large}}^{(l)}$ consisting of a 4-layer MLP, which will be used in the to-be-pruned
 1097 large random Transformers:

$$\mathbf{g}_{\text{large}}(x) = \text{LN}(\mathbf{W}_4 \text{ReLU}(\mathbf{W}_3 \text{ReLU}(\mathbf{W}_2 \text{ReLU}(\mathbf{W}_1 x)))) + x; \tag{29}$$

1100 where $W_1; W_4 \in \mathbb{R}^{m_{\text{large}} \times m_{\text{large}}}$, $W_2; W_3 \in \mathbb{R}^{w_{\text{large}} \times w_{\text{large}}}$, for some $w_{\text{large}}; m_{\text{large}}$.

1101 We are now ready to state the main theorem of this section:

1102 Theorem D.6 (Indistinguishability From a Single Component (Theorem 3.4 restated)). Consider a L -layer Transformer
 1103 T with embedding dimension m , width w and $g^{(k)}(x) = \text{LN}_{\text{C}}(W_2^{(k)} \text{ReLU}(W_1^{(k)}x) + x)$. Suppose $\|W_k\|_2 = O(1)$ for
 1104 every weight matrix W in T . For $\epsilon \in (0, 1)$, consider a large random Transformer T_{large} with $4L$ layers, embedding
 1105 dimension $m_{\text{large}} = O(d \log(d/\epsilon))$, and width $w_{\text{large}} = O(\max\{m; w \log \frac{wmLN}{\epsilon}\})$, and projection functions σ_{large} , whose
 1106 weights are randomly sampled as $W_{i,j} \sim \mathcal{U}(-1; 1)$ for every $W \in T_{\text{large}}$.

1107 Then, with probability $1 - \epsilon$ over the randomness of T_{large} , we can obtain a nonstructural pruning (Definition 2.1) of T_{large} ,
 1108 denoted as T_{large}^0 , which ϵ -approximates T . That is, $\|T_{\text{large}}^0(X) - T(X)\|_2 \leq \epsilon$ with $\|X\|_2 \leq 1; \delta_i \in [N]$,

$$1111 \quad \|T_{\text{large}}^0(X) - T(X)\|_2 \leq \epsilon$$

1112 Moreover, pick any weight matrix W in T_{large} , with probability $1 - \epsilon$, for any smaller Transformer $T_1; T_2$ satisfying same
 1113 conditions as T , we have two pruned Transformers $T_{\text{large},1}; T_{\text{large},2}$ based on T_{large} , such that they coincide on the pruned
 1114 weight of W , and $T_{\text{large},i}$ ϵ_i -approximates T , $\delta_i \in [1; 2g]$.

1115 Proof. We will first introduce some notation. For vectors $x \in \mathbb{R}^a$ and $y \in \mathbb{R}^b$, we will use $x \parallel y$ to denote their concatenation.
 1116 We will use 0^a to denote the all-zero vector with dimension a . We will also assume without loss of generality that $d \geq 7$.

1117 In the following, random network refers to a network whose weights have entries sampled from a uniform distribution, i.e.
 1118 $W_{i,j} \sim \mathcal{U}(-1; 1)$ for every weight W in the random network.

1119 We will first recall Lemma D.7 from (Pensia et al., 2020) which shows that a pruned 2-layer random network can approximate
 1120 a linear function.

1121 Lemma D.7 (Theorem 1 of (Pensia et al., 2020)). Let $W \in \mathbb{R}^{d \times d}$; $\|W\|_2 = O(1)$, then for $\epsilon \in (0, 1)$; g , for a
 1122 random network $g(x) = W_2(W_1x)$ with $W_2 \in \mathbb{R}^{d \times h}$; $W_1 \in \mathbb{R}^{h \times d}$ for hidden dimension $h = O(d \log(\frac{d}{\epsilon \min\{f; g\}}))$, with
 1123 probability $1 - \epsilon$, there exists boolean matrices $M_1; M_2$, such that for any $x \in \mathbb{R}^d$; $\|x\|_2 = O(1)$,

$$1124 \quad \|(M_2 \circ W_2) - (M_1 \circ W_1)x - Wx\|_2 \leq \epsilon$$

1125 where \circ denotes the Hadamard product.

1126 We will use the following helper lemma:

- 1127 1. A pruned 4-layer projection function of a Transformer layer can approximate a 2-layer ReLU network applied to each
 1128 token (Lemma D.8).
- 1129 2. A pruned random Transformer layer can approximate a linear function applied independently to each token (Lemma D.9).
- 1130 3. Two pruned random Transformer layers can approximate a fixed smaller Transformer layer. (Lemma D.12)

1131 We can now prove the theorem.

1132 To show ϵ -approximation, we can prune the large Transformer to approximate the smaller Transformer layer by layer
 1133 by Lemma D.12. The linear head $W^{(\text{head})}$ can be pruned using Lemmas D.9 and D.11, and combined with one layer of the
 1134 Transformer, the linear head of the smaller Transformer can be approximated.

1135 Further, as we only need 2 layers to approximate one layer of the smaller Transformer, for an arbitrary layer l , we can prune
 1136 the layer l of the large Transformer to approximate identity function. This then concludes the proof for indistinguishability
 1137 from single components. \square

1138 ⁷We can always pad dimensions if too small.

1155 D.4.1. HELPER LEMMAS FOR THEOREM D.6

1156 We first show that a pruned 4-layer projection function in a Transformer layer can approximate a 2-layer ReLU network
 1157 applied to each token:

1158 Lemma D.8. Under the condition of Theorem D.6, for any two matrices $W_1 \in \mathbb{R}^{d \times w}; W_2 \in \mathbb{R}^{w \times d}; kW_1k_2; kW_2k_2 = O(1)$,
 1159 for any $\epsilon \in (0; 1)$ and $l \in [4L]$, with probability $1 - \epsilon$, there exists an unstructured pruning $g_{\text{large}}^{(l)}$, satisfying that
 1160 $g_{\text{large}}^{(l)} \in \mathbb{R}^{m \times N}$ with $kX_{:,i}k_2 = O(1); \forall i \in [N]$,

1161
 1162
 1163
$$\forall R \in \mathbb{R}^{(m_{\text{large}} - m) \times N}; g_{\text{large}}^{(l)} \in \mathbb{R}^{m \times N}; \forall X \in \mathbb{R}^{m \times N}; W_2 \text{ReLU}(W_1 X) \approx R g_{\text{large}}^{(l)} X;$$

1164 where $R_{1:m,:}$ denotes the first m rows of a matrix M .

1165
 1166 Proof. Recall the definition of the projection function of a Transformer layer is

1167
 1168
$$g_{\text{large}}^{(l)}(x) = \text{LN} \left(W_4^{(l)} \text{ReLU} \left(W_3^{(l)} \text{ReLU} \left(W_2^{(l)} \text{ReLU} \left(W_1^{(l)} x \right) \right) \right) \right) + x;$$

1169 We will prune the LayerNorm by setting it to the identity. Now we only need to show that there exists boolean matrices
 1170 $M_1; M_2; M_3; M_4$, such that,

1171
 1172
 1173
$$M_4 \left(W_4^{(l)} \text{ReLU} \left(M_3 \left(W_3^{(l)} \text{ReLU} \left(M_2 \left(W_2^{(l)} \text{ReLU} \left(M_1 \left(W_1^{(l)} X \right) \right) \right) \right) \right) \right) \right) \approx R g_{\text{large}}^{(l)} X;$$

1174
 1175 We can first choose

1176
 1177
 1178
$$(M_1)_{:, (m+1) : \dots : m_{\text{large}}} = 0; (M_4)_{(m+1) : \dots : m_{\text{large}}, :} = 0;$$

1179
 1180 Then by Lemma D.7, there exists boolean matrices $M_2; M_3; M_4$ satisfying previous constraint, such that,

1181
 1182
 1183
$$(M_2 \ W_2^{(l)}) \text{ReLU} \left(M_1 \ W_1^{(l)} \right) X \approx \begin{matrix} 2 & 3 \\ W_1 & \\ 4 & | & 5 \\ \hline & & 4 \end{matrix} X;$$

1184
 1185 This then concludes the proof. □

1186
 1187 Based on the above lemma, we can prove that a pruned Transformer layer can approximate a linear function applied
 1188 independently to each token.

1189 Lemma D.9. Under the conditions in Theorem D.6, for any matrix $W \in \mathbb{R}^{m \times m}; kWk_2 = O(1)$, $\epsilon \in (0; 1)$ and
 1190 $l \in [4L]$, with probability $1 - \epsilon$, there exists an unstructured pruning $T_{\text{large}}^{(l)}$, satisfying that $\forall X \in \mathbb{R}^{m \times N}$ with
 1191 $kX_{:,i}k_2 = O(1); \forall i \in [N]$, we have

1192
 1193
$$\forall R \in \mathbb{R}^{(m_{\text{large}} - m) \times N}; T_{\text{large}}^{(l)} \in \mathbb{R}^{m \times N}; \forall X \in \mathbb{R}^{m \times N}; WX \approx R T_{\text{large}}^{(l)} X;$$

1194
 1195 Proof. Recall that given an input X^0 , a Transformer layer computes $g_{\text{large}}^{(l)}(X^0) = g_{\text{large}}^{(l)} \left(\text{LN} \left(W_V^{(l)} X^0 \text{Attn}(X^0) \right) \right) + X^0$,

1196
 1197 where $\text{Attn}(X^0) := C \frac{(W_K^{(l)} X^0)^T (W_Q^{(l)} X^0)}{d_a}$ computes the attention pattern. Lemma D.8 already shows that $g_{\text{large}}^{(l)}$ can
 1198 approximate a linear transformation; it remains to show that the linear transformation can compute WX .

1210 We can first choose two matrices $W_1 \in \mathbb{R}^{m \times m}$; $W_2 \in \mathbb{R}^{m \times w}$ satisfying that

$$1211 \quad W_1 = [I_m; I_m; 0^{m \times (w-2m)}];$$

$$1212 \quad W_2 = [W; W; 0^{m \times (w-2m)}]$$

1215 Then we have that $\|W_1\|_2, \|W_2\|_2 = O(1)$ and $W_2 \text{ReLU}(W_1 X) = WX$. We can then turn off the LayerNorm after the
 1216 attention module and prune W_V to be 0, which effectively removes the effect of attention and rely solely on the residual link.
 1217 The proof can now be completed by applying Lemma D.8. \square

1219 We will then show that two pruned Transformer layers can approximate a fixed smaller Transformer layer. The key technical
 1220 difficulty is approximating the attention module and bounding the error of the approximation after LayerNorm. We will first
 1221 show a lemma showing the Lipschitzness of the LayerNorm (with cutoff at some constant

1222 Lemma D.10. For LayerNorm function defined as $\text{LN}(x) = \frac{P \cdot x}{\max\{P, \|x\|_2\}}$; $x \in \mathbb{R}^m$, there exists constant C_1 depending
 1223 on C , such that for any $x, y \in \mathbb{R}^m$, it holds that,

$$1224 \quad \|\text{LN}(x) - \text{LN}(y)\|_2 \leq C_1 \|x - y\|_2$$

1228 Proof. We will proceed by a case analysis:

- 1230 1. If $\|x\|_2, \|y\|_2 \leq C$, then $\|\text{LN}(x) - \text{LN}(y)\|_2 = \frac{\|P \cdot x - P \cdot y\|_2}{C} \leq \frac{1}{C} \|x - y\|_2$.
- 1233 2. If $\|x\|_2, \|y\|_2 > C$, then $\|\text{LN}(x) - \text{LN}(y)\|_2 = \frac{\|P \cdot x - P \cdot y\|_2}{\|x\|_2 \|y\|_2} + 1 \leq \frac{\|x - y\|_2}{\|x\|_2 \|y\|_2} + \frac{2}{C} \|x - y\|_2$.
- 1236 3. If $\|x\|_2 < C$ and $\|y\|_2 > C$, then $\|\text{LN}(x) - \text{LN}(y)\|_2 = \frac{\|P \cdot x - P \cdot y\|_2}{\|x\|_2 \|y\|_2} + \frac{\|x\|_2}{\|y\|_2} \leq \frac{\|x - y\|_2}{\|x\|_2 \|y\|_2} + \frac{2}{C} \|x - y\|_2$.

1238 The cases exhaust all possibilities, thus the proof is completed. \square

1240 We also need to show there exists a pruning of the value matrix in such that it has eigenvalues with magnitude $\geq \frac{1}{2}$.

1241 Lemma D.11. For a matrix $W \in \mathbb{R}^{w_{\text{large}} \times w_{\text{large}}}$, with probability at least $1 - \epsilon$, there exists a pruning W^0 , named W^0 ,
 1242 such that all the nonzero entries is contained in a d submatrix of W^0 that satisfies that (1) all its eigenvalues are within
 1243 $(\frac{1}{2}, 1)$, (2) the index of row specifying the submatrix and the index of column specifying the submatrix are disjoint.

1246 Proof. As $w_{\text{large}} = \Theta(m \log(d))$, hence we can split W into $\frac{w_{\text{large}}}{d} = \Theta(m)$ blocks, each with
 1247 width at least $\Theta(\frac{w_{\text{large}}}{d})$. Within each block, with probability $1 - \frac{\epsilon}{\Theta(m)}$, there exists at least one entry that has value at least
 1248 $\frac{1}{2}$. We can then choose disjoint entries in W that are all at least $\frac{1}{2}$, indexed with $(a_i, b_i)_{i \in [d]}$ where $a_i < a_j$ and $b_i < b_j$
 1249 for $i < j$. We can then prune all other entries to zero. Consider the submatrix defined by entries for a 2×2 submatrix and
 1250 $b_2 \times b_1$ submatrix. Then, this submatrix will be diagonal and contains eigenvalues within $(\frac{1}{2}, 1)$. Further, $a_i \in [2m]$ and $b_i \in [2m]$
 1251 must be disjoint because $d \leq m_{\text{large}} = 2e < b_i$. The proof is then completed. \square

1253 Next, we show that two random Transformer layers can be pruned to approximate a given Transformer layer.

1254 Lemma D.12. Under the condition of Theorem 3.4, for any matrix $M \in \mathbb{R}^{d \times d}$; $\|M\|_2 = O(1)$, $\lambda \in (0, 1)$ and $t \in [L]$, for
 1255 any $l \in [L]$, with probability $1 - \epsilon$, there exists an unstructured pruning $T_{\text{large}}^{(t)} \cdot T_{\text{large}}^{(t+1)}$, named $T_{\text{large}}^{(t)}$; $T_{\text{large}}^{(t+1)}$, satisfying
 1256 that $\|X \in \mathbb{R}^{d \times N}$ with $\|X_{:,i}\|_2 = O(1)$; $\forall i \in [N]$,

$$1259 \quad \|R \in \mathbb{R}^{(m_{\text{large}} - m) \times N}\|_2 \leq \epsilon; \quad \|T_{\text{large}}^{(t+1) \circ} \cdot T_{\text{large}}^{(t) \circ} [X_{:,i} \cdot R_{:,i}]_{i \in [N]} - T_{\text{large}}^{(t+1)}(X)_{:,i}\|_2 \leq \epsilon$$

1262 Proof. We will prune the larger transformer in the following order.

1263 ⁸ $O(\cdot)$ hides absolute constants arising from the change of basis in the logarithm.

1265 1. We will prune $W_V^{(t+1)}$ according to Lemma D.11 and name the pruned matrix $W_V^{(t+1)^\circ}$. By Lemma D.11, all the nonzero
 1266 entries is contained in a d submatrix of W^0 that satisfies that all its eigenvalues are within $(\frac{1}{2}, 1)$. We will prune
 1267 $W_V^{(t+1)}$ in this way, named $W_V^{(t+1)^\circ}$ and assume WLOG the submatrix is the one specified by row d and column
 1268 $d+1 :: : 2d$ and name the submatrix M .

1270 2. We will then prune $T_{large}^{(t)}$ according to Lemma D.9 to output approximation of $X_{:,i} = W^{-1} P_{\gamma} W_V^{(l)} X_{:,i}$
 1271 $A_{:,i}$ for some vectors $A_{:,i}$. As W is defined as the submatrix pruned by $W_V^{(t+1)}$, it holds that
 1272 $W_V^{(t+1)^\circ} X_{:,i} = W^{-1} W_V^{(l)} X_{:,i} = A_{:,i} = P_{\gamma} W_V^{(l)} X_{:,i} + O^{m_{large}-m}$.

1275 3. We will then prune $W_K^{(t+1)}$ and $W_Q^{(t+1)}$ according to Lemma D.7 to approximate attention patterns. We will choose
 1276 boolean matrix $M_K; M_Q$ such that for any $x \in \mathbb{R}^d$ and $a \in \mathbb{R}^{m_{large}-m}$,

$$\| (M_K W_K^{(t+1)})^T (M_Q W_Q^{(t+1)} (x - a)) - (W_K^{(l)})^T W_Q^l x \|_2 = O^{m_{large}-m} \|x\|_2$$

1280 We can then have that the attention pattern for the large transformer at layer t can approximate the small one. That is,
 1281 for any $x \in \mathbb{R}^d; \|x\|_2 = O(1)$ and $a \in \mathbb{R}^{m_{large}-m}$,

$$(x - a)^T (M_K W_K^{(t+1)})^T (M_Q W_Q^{(t+1)} (x - a)) - x^T (W_K^{(l)})^T W_Q^l x = O(\epsilon)$$

1285 Combined with previous approximation $W_V^{(t+1)^\circ} X_{:,i} = W^{-1} W_V^{(l)} X_{:,i} = A_{:,i}$ and the Lipschitzness of the
 1286 LayerNorm, we have that the r th dimensions of the output after LayerNorm of the large Transformer at layer
 1287 can ϵ -approximate the output after LayerNorm of the smaller Transformer at layer

1290 4. We will finally prune the MLP in the projection function of $T_{large}^{(t+1)}$ to approximate $P_{\gamma} f^{(l)}$ with $f^{(l)}$ being the MLP in
 1291 the projection function of the projection function $T(f^{(l)})$.

1293 The proof is then complete. □

1320 E. Experiments

1321 E.1. Training Details

1323 For Figure 1, we train 2-layer standard GPT-Dyck_{2,4} with sequence length no longer than 200. For (a), we train with
 1324 hidden dimension and network width 200 and learning rate 3e-4. For (b); (c); (d), we train with hidden dimension and FFN
 1325 width 50 and learning rate 3e-3.

1326 For Figure 2, for (a), we train 1-layer transformer without residual link, FFN and the final LayerNorm before the linear
 1327 head. The hidden dimensions and FFN widths are 500. For (a), we train the network with learning rate 1e-2 and for
 1328 (b); (c); (d) we train the network with learning rate 3e-3.

1330 let \mathbf{o}_t denote the one-hot embedding where $\mathbf{o}_t[t] = 1$,

$$1332 \mathbf{e}_{t;d} = \mathbf{o}_t \oplus \mathbf{0}_{D+d}; \quad (\text{Type 1})$$

$$1333 \mathbf{e}_{t;d} = \mathbf{o}_t \oplus [\cos(\theta_d); \sin(\theta_d)]; \quad (\text{Type 2})$$

$$1334 \theta_d = \arctan(d/(D+2-d));$$

$$1336 \mathbf{e}_{t;d} = \mathbf{o}_t \oplus \mathbf{0}_d; \quad (\text{Type 3})$$

1338 Operator \oplus means the concatenation of two vectors. Equation (Type 1) is the standard one-hot embedding for
 1339 and Equation (Type 3) is the concatenation of one-hot embedding of types and depths. Finally, Equation (Type 2) is the
 1340 embedding constructed in Yao et al. (2021).

1342 E.2. Guiding The Transformer To Learn Balanced Attention

Figure 6: Relationship Between Balance Violation and Length Generalization. Accuracy from Transformers with minimal first layer with embedding Type 1, using both standard training and contrastive regularization (Equation (30)). Standard training leads to high balance violations which negatively correlate with length generalization performance. Contrastive regularization helps reduce the balance violation and improve the length generalization performance.

1357 In our experiments, we observe that although models learned via standard training that can generalize well in distribution,
 1358 the length generalization performance is far from optimal. This implies that the models are not finding the correct algorithm
 1359 for parsing Dyck when learning from finite samples. A natural question is: can we guide Transformers towards correct
 1360 algorithms, as measured by better generalization on longer Dyck sequences?

1362 In the following, we measure length generalization performance by testing the accuracy of the model on Dyck sequences
 1363 with length randomly sampled from 400 to 500, which approximately correspond to 16 times the length of the training
 1364 sequences. We will show generalization can be improved by regularizing the attentions to be more balanced, inspired by
 1365 results in Section 3.

1367 Balance violation negatively correlates with length generalization accuracy. We denote the balance violation of a
 1368 Transformer as $\mathbf{E}_{d;d^0;ij}[\mathbf{S}_{d;d^0;ij} = \mathbf{P}_{d;ij}]$ for $\mathbf{S}; \mathbf{P}$ defined in Equations (5) and (7). Theorem 3.2 predicts that for models
 1369 with a minimal first layer, perfect length generalization requires to be zero. Beyond such idealized condition, it is natural
 1370 to ask whether a small yet positive correlation with length generalization accuracy in practice. Our results show a moderate
 1371 correlation ($\rho = 0.38$ SpearmanR with p-value 0.014) based on over 40 random initializations (Figure 6).

1372 Given the correlation, we design a contrastive training objective to reduce the balance violation, which ideally would lead to
 1373 improved length generalization. Specifically, let \mathbf{u} denote a prefix of nested pairs of brackets of form $\mathbf{u} \in \mathcal{U}([D])$, and

let $T(s; j, p_r, s)$ denote the logits for s when T takes as input the concatenation of p_r and s . We define the contrastive regularization $R_{\text{contrastive}}(s)$ as the mean squared error between the logits $T(s; j, p_r, s)$ and $T(s; j, p_r, s)$, taking expectation over p_r :

$$E_{p_r \sim U(\{D\}); p_r} \|T(s; j, p_r, s) - T(s; j, p_r, s)\|_F^2 \quad (30)$$

Following the same intuition as in the proof of Theorem 3.2, if the model can perfectly length-generalize, then the contrastive loss will be zero. We then train the model with contrastive loss and observe that the balance violation is reduced and the length generalization performance is improved (Figure 6).

E.3. Additional Results on Dyck Prefix

In the experiment presented in the main text, we perform experiments on complete Dyck sequences, which is a special case of Dyck prefixes. In this section, we present additional experiments on Dyck prefixes.

Attention Patterns We first perform experiments on attention patterns. The qualitative results are shown in Figures 7 and 9. We can observe that the attention patterns are still diverse and do not commonly show stack-like patterns. We also calculate the attention variation⁹, and find that the attention variation is 0.34, based on 30 models with a minimal first layer and different random seeds. In contrast, for models with a standard first layer and without position encodings, the attention variation is surprisingly high, reaching 1.51. The high value is caused by the large distance between attention patterns like Figure 7 (c) and (d); that is, between patterns that attend more to the current positions, and patterns that attend more heavily to the initial position. The difference is even increased when we consider longer sequence (Figure 8). Similarly, the variation is also high for models with linear position embedding, reaching 1.02. This shows that the attention patterns are still diverse and do not commonly show stack-like patterns.

(a) With Position
Embedding

(b) With Position
Embedding

(c) Without Position
Embedding

(d) Without Position
Embedding

Figure 7: Second-layer attention patterns of two-layer Transformers on Dyck Prefix Models for (a),(b) are under the same setup but different random seeds; similarly for (c),(d). All models reach 97% accuracy (defined in ??). In the heatmap, darker color indicates larger value. As we can observe, the attention patterns still show much variance.

Balanced Violations We also test the relationship with the balance violation with length generalization on Dyck prefixes, similar to Figure 6. We observe that although the negative correlation is not presented as in the case of Dyck sequences, contrastive regularization still helps reduce the balance violation and significantly improve the length generalization performance. This shows that for Dyck prefixes, while the balance violation may not be predictive of the length generalization performance, it is still possible to reduce the balance violation and improve the length generalization performance. The results are shown in Figure 10.

⁹Recall from ?? that the attention variation between two attention patterns $A_1, A_2 \in \mathbb{R}^{N \times N}$ is defined as $\text{Variation}(A_1; A_2) = \frac{1}{2} \|A_1 - A_2\|_F^2$:

